Reshaping the Retail Industry using AI vision

- **Due:** *11:59 pm 22/5/2024 (Wednesday of Week 12)*
- **Contributes** 50% of your final result
- **Group Assignment – Group of 3-4 students**

Summary

AI is revolutionizing the retail industry by offering innovative solutions to streamline traditional supply chain processes, ranging from product distribution to inventory management. In response to the rise of online shopping, the retail sector has swiftly embraced AI technologies at the physical store level to automate labor-intensive tasks through the utilization of computer algorithms and robotic systems. The integration of AI in retail holds immense potential, with numerous applications poised for widespread adoption in the near future. One notable application is the use of computer vision techniques, which enables retailers to optimize store operations and enhance customer experiences. For instance, computer vision allows for the automated auditing of product placements and monitoring of in-store sales activities by analyzing shelf images. Moreover, customers can leverage this technology to access product information by simply capturing images of the items they are interested in. In this project, the focus is on applying computer vision techniques to address two significant challenges in retail execution: the detection of missing items on store shelves (out-of-stock detection) and ensuring effective product management.

Out-of-stock detection is a critical aspect of retail operations, and AI-powered computer vision systems play a pivotal role in efficiently identifying and rectifying instances where products are unavailable for customers. By automating this process, retailers can reduce the likelihood of missed sales opportunities and enhance customer satisfaction by ensuring that popular products are consistently in stock.

Effective product management is essential for ensuring precise and well-organized product arrangement, aligning with planogram compliance. In this project's basic version, the primary focus is on evaluating the following cases on product placement:

- Are the shelves filled properly with item for visual appeal or are the unnecessary gaps?
- Are shelves neatly organized on the basis of the product type/category?

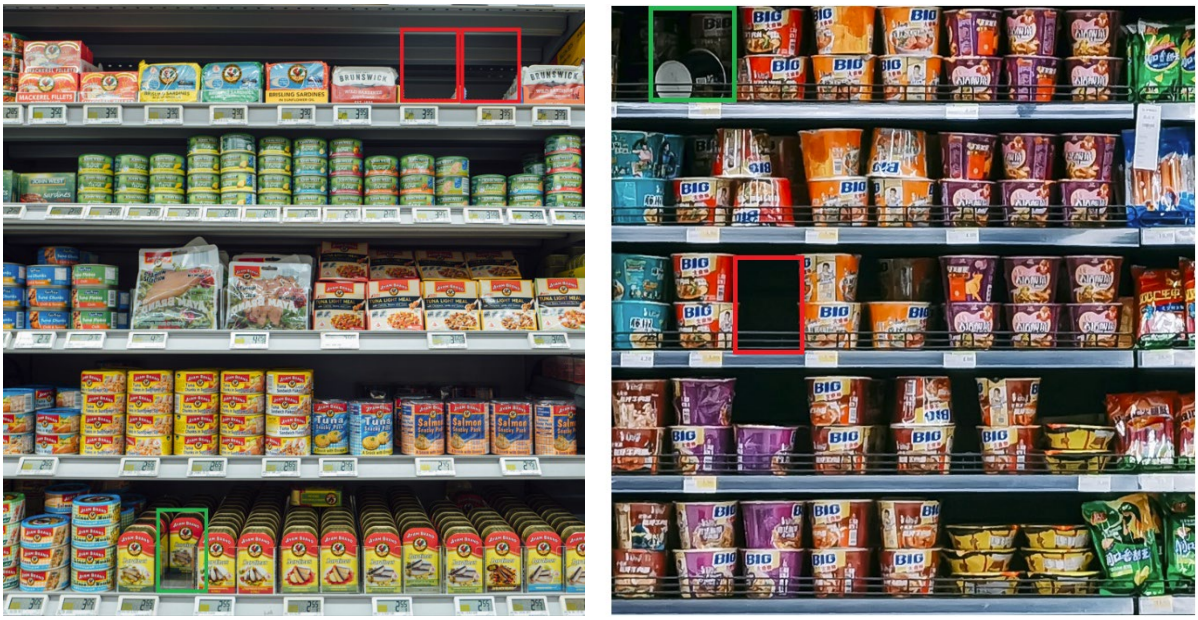

Figure 1: The red boxes show examples of out-of-stock detection while the green boxes represent examples where planogram compliance has not met expectations.

Constraints:

Object Detection System

To effectively implement these applications, a fundamental challenge is to develop a robust object detection system tailored to specific regions of interest. For example, in the context of detecting missing items, the

region of interest would include identifying empty shelves. Conversely, when it comes to ensuring product management, the focus is on locating shelves containing disorganized items and identifying cases where items are placed with unnecessary gaps.

In this project, you have to implement and compare the performance of two distinct object detection pipelines. These pipelines include **one-class object detection** and **multi-class object detection**. The first pipeline involves training separate one-class object detection models tailored to specific scenarios: out-of-stock detection and product placement evaluation. This approach allows us to focus on training models to recognize and distinguish objects relevant to each scenario independently. The second pipeline centers around training a multi-class object detection model capable of simultaneously detecting both scenarios. This approach involves creating a unified model that can identify and classify objects related to both out-of-stock detection and product placement evaluation within the same framework.

By implementing and evaluating these two pipelines, you will assess their respective strengths and weaknesses, ultimately determining which approach yields superior performance for the specific challenges of this project.

Retail Product Detection Dataset

Please note that you have to collect and annotate your own training and testing sets for this project. Ensure there is no overlap between the images used for training and testing to maintain the integrity of the evaluation process. You can utilize a similarity checker to identify and exclude duplicate data. Allocate a minimum of 300 images for each scenario. Use an 80% split for training and a 20% split for testing. To annotate the images accurately, leverage annotation tools such as LabelMe or other applications supporting bounding box annotations. Below you'll find data on supermarket shelves that you can use, but you'll need to annotate the regions of interest yourself.

Supermarket Shelves dataset
https://www.kaggle.com/datasets/humansintheloop/supermarket-shelves-dataset

RP2K: A Large-Scale Retail Product Dataset for Fine-Grained Image Classification
https://www.pinlandata.com/rp2k_dataset/

System requirements:

Regarding the object detection module, you should take advantage of existing libraries such as PyTorch, Tensorflow, Keras, Theano, etc. At the very least, you should re-implement the object detection technique that will be taught in the lab session. After this attempt, you are expected to understand the methodology behind object detection and know how to form an object detection model using deep learning. Next, you should be able to adapt the code to work with retail execution using your own collected and annotated dataset.

- **Basic version:**
  - The basic version involves implementing the two previously mentioned approaches: One-class and multi-class object detection.
  - The graphical user interface (GUI) should encompass the following features:
    - Object detection capabilities for both image and video data.
    - Display of bounding boxes highlighting the two scenarios, namely out-of-stock detection and planogram compliance adherence, for both image and video data.
- **Extension 2**: In real-world applications, not all shelves are consistently captured with a frontal angle. Various scenarios involve cameras mounted at different angles, introducing viewpoint variation that can impact the effectiveness of object detection models trained for assessing missing objects and ensuring proper product placement. In this extension, students are challenged to collect their own dataset featuring shelves from diverse angles and propose innovative techniques to mitigate the challenges associated with viewpoint variation. Substantial research efforts are expected to devise effective solutions for this problem, ensuring the robust performance of object detection models in scenarios with varying perspectives.
- **Extension 3**: Another key challenge, involves leveraging computer vision to verify whether the actual store layout aligns with the prescribed planograms. This includes verifying the correct placement of products on shelves, adhering to specified facings, and ensuring that the visual

merchandising aligns with the intended marketing strategy. Through AI-driven computer vision, retailers can maintain a visually appealing and strategically organized store layout, thereby maximizing sales and enhancing the overall shopping experience. As part of this extension, students are tasked with designing their own planogram and composing a new dataset that corresponds to the created planogram to facilitate planogram matching. The dataset should authentically reflect real-world scenarios, showcasing instances of item misplacement that deviate from the designed planogram, as well as examples where items adhere precisely to the planogram. The trained model is expected to effectively detect instances of misplaced items that do not adhere to the planogram, providing valuable insights for optimizing store layout and ensuring planogram compliance.

Project requirements

- Source code maintained on Git based VCS (Github/Bitbucket/GitLab/…). You must provide read-only access to the tutor/lecturer
- Running illustrative demo of a working prototype (please refer to **Marking Scheme** for details on functionality that needs to be implemented)
- Project report (8-10 pages) that includes the following sections
    - Cover Page (with team details) and a Table of Contents (TOC)
    - Introduction
    - Overall system architecture
    - Data collection and annotation,
    - Implemented machine learning techniques,
    - Scenarios/examples to demonstrate how the system works,
    - Some critical analysis of the implementation,
    - Practical application description and
    - Summary/Conclusion.
    - Presentation + demo video link (10 minutes duration)

Marking Scheme

Requirements	***Mark***
Task 1: Data collection and annotation. Implement the one-class object detection.	25
Task 2: Implement multi-class object detection. Conduct a comparative analysis of model performances between one-class and multi-class object detection. Deliver a critical analysis with in-depth findings.	25
GUI: Display the real-time predictions of bounding boxes using two different data modalities: images and videos.	10
Project Report	10
Project Presentation (Video)	10
	80
Research Component (can be done by the whole team, a sub-team, or an individual) There are many potential extensions in this project. **Extension 1** and **Extension 2** are two examples but there are many more. Choose one and get your tutor's approval then complete it very well. You can get up to 40% for this component and your Assignment will get up to 120%	Up to 40
	120/100
You need to follow good programming practice (e.g., well-designed, well-structured codes with clear and helpful comments). Failure to do so get penalty.	**Up to -20**
You need to demonstrate the progress you make every week to your tutor. That is, if your tutor approach you and ask for the progress, you have to be able to show the tutor the progress you have made in comparison to the previous week. Failure to do so will get a penalty.	**Up to -50**

NOTE:

- Individual marks will be proportionally adjusted based on each team member's overall contribution to the project as indicated in the '*Who did what*' declaration.
- You must also provide to shlee@swinburne.edu.my read only access to your git repository within 1 week of forming teams.

Submission

- You must upload your work to Canvas by 11:59pm on 22/5/2024 (Wednesday). Create a single zip file with your code and a working version of your system. Standard late penalties apply - 10% for each day late, more than 5 days late is 0%.
- The video (10 minute duration) link should be stated in the project report.